

並列オブジェクト指向トータルアーキテクチャA-NET

2L-5

— ホストマシンソフトウェアの実装 —

廣田守 吉永努 馬場敬信

宇都宮大学工学部*

1. はじめに

並列計算機のユーザにとって、使いやすいユーザインタフェースが提供されることは、極めて重要である。

我々は、並列オブジェクト指向トータルアーキテクチャA-NETプロジェクト [1, 2] においてマルチコンピュータを開発した。この多数のノードを制御するためには、ホストマシンとその上のソフトウェアが重要な役割を果たす。

本研究の目的は、使いやすいユーザインタフェースの実現を目的としてホスト上のソフトウェアシステムを構築することである。

以下、本稿では、A-NETマルチコンピュータ（以後、A-NET計算機）のホストプログラムの設計方針、実現、主要な処理の流れについて述べる。

2. ホストプログラムの設計方針

ホストプログラムを実現する際の設計方針は次の通りである。

- ホストをオブジェクトと同じ概念で取り扱う。従って、ユーザがホストにメッセージを送信する場合には、A-NETLで定義されたメッセージ式で入力する。また、A-NET計算機からもメッセージにより処理の依頼を行なう。
- A-NETLプログラミング支援システム（以後、APSS）及び、A-NETLデバッガ（以後、ADEB）[3]上からの起動を可能とする。これにより、GUIを用いた操作が可能になり、ホストプログラムを介してA-NET計算機との通信が可能となる。

3. ホストプログラムの実現

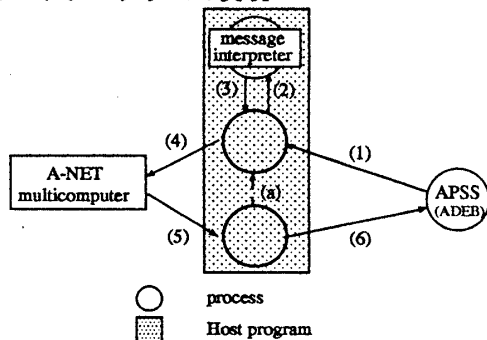


図1 A-NET計算機・APSS(ADEB)・ホストプログラムの相互関係

*A Parallel Object-Oriented Total Architecture A-NET - Implementation of the host machine software -, Mamoru Hirota, Tsutomu Yoshinaga and Takanobu Baba, Utsunomiya University.

処理系におけるホストプログラムの位置付けを明確にするためにA-NET計算機、APSS(ADEB)、ホストプログラムの相互関係を図1に示す。

なお、A-NET計算機の各ノードは要素プロセッサ(PE)とルータから成る。

図1中の番号に沿ってデータの流れを説明する。(1)APSSからホストプログラムにメッセージ式を入力。(2)ホストプログラムが受けとったメッセージ式をメッセージインタプリタに入力。(3)翻訳されたメッセージ式をホストプログラムが受けとる。(4)そのメッセージをA-NET計算機に送信。(5)A-NET計算機からのメッセージをホストプログラムが受信。(6)解釈されたメッセージを結果として出力。(a)の破線は、ユーザプログラムをロードする際、2つのプロセスが同期をとるためのデータの流れである。

ここで、ユーザ(APSS・ADEB)からホストへ送るメッセージのセレクトを表1に示す。

表1 ユーザ(APSS・ADEB)からホストへ送信されるセレクトの内容

セレクト	内容
init	初期化、マイクロプログラム・ローカルOSのロード
halt	全ノードに停止命令を送信、ホストの終了
reset	リセット信号を全ノードに送信、ローカルOSを再ロード
load	ロードモジュールのロード
dump	A-NET計算機のローカルメモリのダンプ出力
wrlm	A-NET計算機のローカルメモリへのデータの書き込み
sh	UNIX土でのUNIXコマンド実行

ユーザがメッセージ式を入力すると、それをメッセージインタプリタが翻訳し、その結果をA-NET計算機に送信する。図1の(1)(2)(3)(4)に対応する。

次に、A-NET計算機からホストに送信されるメッセージのセレクトを表2に示す。

表2 A-NET計算機からホストへ送信されるセレクトの内容

セレクト	内容
haltsel	対象ノードのPEの停止
dmasel	ローカルOSのロードの要求
dmafinishsel	ローカルOSのロードの終了
dmaacksel	ユーザプログラムのロードの要求
printsel	画面出力
eventrapsel	評価用データ
mdebssel	デバック用データ

ホストはA-NET計算機からのメッセージを受信し、受信したメッセージのセレクトによって各処理を行なう。図1の(5)(6)(a)に対応する

A-NET計算機へのアクセスは、A-NET VME I/F デバイスドライバにより行なっている。

3. 主要な処理の流れ

これまでに開発した A-NET 計算機モニタによって、低レベルでの操作は可能であるが、ユーザプログラム実行の操作性、現在開発中の ADEB を考慮に入れると、ハードウェアを意識しないで容易に操作できるようなホストプログラムが必要である。そのことを踏まえて、ユーザの負担を軽減させるためになるべく処理の自動化を図っている。

ホストの主要な機能にはシステムスタート及びユーザプログラムロードがある。以下にその処理を説明する。

● システムスタート

システムスタートの処理は、大きく分けてマイクロプログラムのロード及び、ローカル OS のロードに分れる。マイクロプログラムには、通常実行用、実行評価用、デバッグ用の 3 種類が存在する。これらの選択は、ホストプログラム起動時のオプションにより決定される。

A-NET 計算機の電源を投入後、ホストプログラムを起動させる。図 2 に初期化メッセージ式 (HOST init) を入力した際の処理内容を示す。

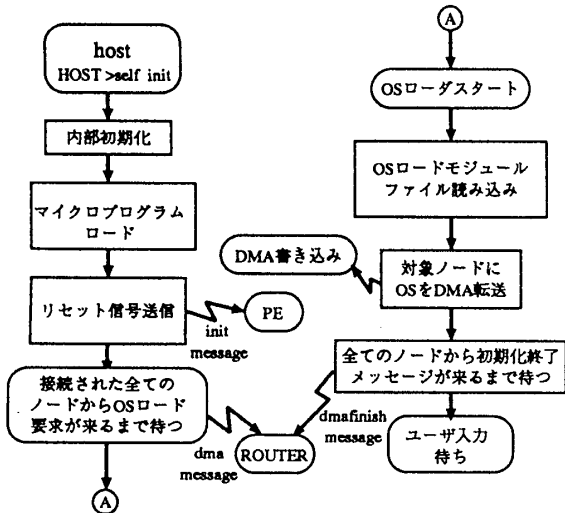


図 2 システムスタート

● ユーザプログラムのロード

ロードの際、ノーマルオブジェクトは対象ノード、インデックストオブジェクト及びクラスオブジェクト [1] は全ノードにブロードキャストされる。

ホストプログラムのプロセスは、A-NET 計算機に対して送信と受信の 2 つのプロセスに分れている。従って、連続してオブジェクトをロードする場合には同期をとる必要があり、プロセス間通信によってこれに対応している。

図 3 にユーザプログラムのロード時の処理を示す。

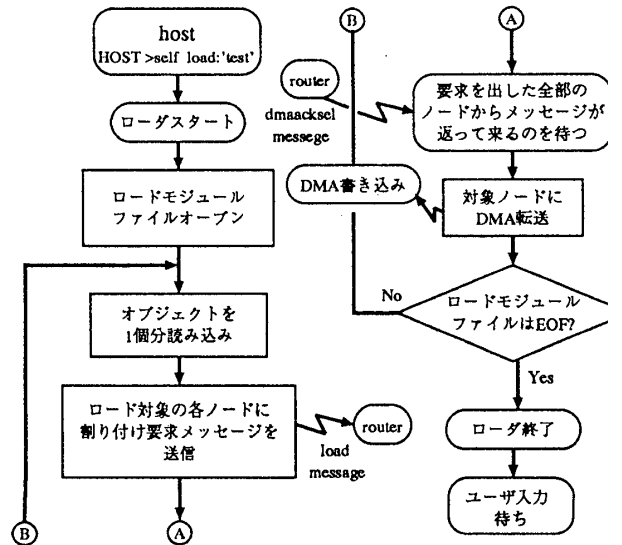


図 3 ユーザプログラムロード

A-NET 計算機のノードからメッセージがホストに送信されずホストがそれを待ち続けた場合、ホストはタイムアウト機能によりユーザにエラーを出力し、それ以後はそのノードを無視する。

5. おわりに

本研究では、A-NET 計算機のホストプログラムの設計方針を立て、VME I/F デバイスドライバを用いて構築した。

また、実際にホストプログラムを APSS 上から起動し、Life Game などのユーザプログラムを実行させ正しい実行結果を得ている。

ADEB 上からの使用についても、現在までに開発されているブレイクポイント機能などは、ホストプログラムを介し A-NET 計算機と情報交換しながら完動している。

現在、A-NET 計算機で実行評価をとるためのマイクロプログラムを開発している。それを受けて今後、ホストプログラムも評価モードに対応させる必要がある。

謝辞

本研究は、一部文部省科学研究費、一般(C)07680334、奨励(A)07780225、および電気通信普及財団の援助による。

参考文献

- [1] 馬場敬信, 吉永努, “並列オブジェクト指向トータルアーキテクチャ A-NET における言語とアーキテクチャの統合”, 信学会論文誌 Vol.J75-D-I No.8 pp.563-574(1992)
- [2] 吉永努, 馬場敬信, “並列オブジェクト指向トータルアーキテクチャ A-NET - マルチコンピュータ開発と言語実装の現状 -”, 情報処理学会 第 52 回全国大会予稿集 2L-3(1996)
- [3] T. Baba, T. Yoshinaga and T. Furuta, “Programming and Debugging for Massive Parallelism: Case for a Parallel Object-Oriented Language A-NETL”, Proc. OBPDC '95(1995)